View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Open University of the Netherlands Research Portal

PROFIL: a method for the development of multimedia courseware

Citation for published version (APA):

Koper, R. (1995). PROFIL: a method for the development of multimedia courseware. *British Journal of Educational Technology*, *26*(2), 94-108. https://doi.org/10.1111/j.1467-8535.1995.tb00127.x

DOI:
10.1111/j.1467-8535.1995.tb00127.x

Document status and date:
Published: 01/05/1995

Document Version:
Peer reviewed version

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:

https://www.ou.nl/taverne-agreement

Take down policy
If you believe that this document breaches copyright please contact us at:

pure-support@ou.nl

providing details and we will investigate your claim.

Downloaded from https://research.ou.nl/ on date: 26 Nov. 2020

Open Universiteit
www.ou.nl

PROFIL: a method for the development of multimedia courseware[1]

E.J.R. Koper
Centre for Educational Technology
Open university of the Netherlands
Valkenburgerweg 167
6419 AT Heerlen
The Netherlands
Tel.: ...31-45762657
Fax: ...31-45762800

Abstract

This article describes a dedicated method for the design of multimedia courseware, called PROFIL. The method integrates instructional design methods and techniques with software engineering methods and techniques. Furthermore it integrates media selection methods in the design methodology and it takes account of the design of courses of which courseware is a only a part. In six phases (preliminary investigation, definition, script, technical realisation, implementation and exploitation) a program is designed and produced. The essence of the method is that a distinction is made between functional objects and the implementation of the functional objects in the available media and that the same design process is repeated a few times at different aggregation levels.

1.Introduction

This article deals with the problem we had at the Dutch Open university (Ou) with the design of educational software. The Ou - founded in 1984 - provides higher distance education for individual learners who study primarily at home at their own pace. For specific facilities and tutoring students can go to one of the 24 study centres; 18 in the Netherlands and 6 in Belgium. From the beginning, educational software was considered as one of the media through which the education could be delivered. At this moment about 135 different courseware packages are used in the different courses of the seven faculties of which more than 60% is developed at the university itself (at the Centre for Educational Technology). About 30% of the programs are educational simulations/games, 30% instrumental programs, 30% tutorials, and 10% tests and drills. Twenty programs are multimedial (interactive video, CD-I or CD-ROM). The courseware is developed in multidisciplinary teams of: educational technologists, content matter specialists, programmers, graphical designers and an audio-visual production team (for multimedia productions).

The problem we had several years ago, was that there wasn't a suitable method for courseware development which met our demands. First of all we wanted a method that fully integrated instructional design methods with software engineering techniques. Some developers consider the production of courseware as a special case of software development which can be managed with usual software development methodologies, such as SDM (Eilers, 1979), Yourdon (Yourdon and Constatine, 1978, Coad & Yourdon, 1991). Others consider it as a special case of instructional design. In fact it is both and in our view this does not simply mean that the first phase of the development process is based on instructional design techniques, followed by software development techniques (e.g. Alessi & Trollip, 1985, Hartemink, 1987, Roblyer, 1988, Wager & Gagné, 1988). Special attention has to be given to the integration of the two approaches. Second, we wanted a method which integrates media selection methods in the design process. The systematic selection and justification of media for the implementation of didactic functions is in our opinion a major factor for the quality of courseware. Furthermore we think that courseware

[1] Published in the British Journal of Educational Technology, volume 26, number 2, May 1995.

should be developed as an integrated part of courses which consist of a media mix (such as books, persons and audio-visual media) and not only of a computer program and some additional support. This means that the development methodology must support the idea of designing the computer program in this larger context.
Third, we wanted a method which could be used for every type of program we develop: tutorials, games, simulations, instrumental programs, tests, multimedia practicals, etc. Some methods are suitable for some types of programs, but not for others. The methods described by Bork (1984, 1986) and by Burk (1982) for instance are specifically suited for tutorials, but less for educational simulations. The same is true for programming techniques: some methods fit an object-oriented or procedural approach of programming, but are not well suited for declarative or functional techniques.
Fourth, we didn't want a waterfall approach, but more a spiral or incremental approach of development. In every design stage the whole program is thought through in all its aspects, but at a different aggregation level. This is especially necessary in the preliminary stages for planning en testing purposes. Prototyping and tests with students are aspects of such an approach.
The last major demand was that the development method should support teamwork and working in projects. The method must recognise the problem of communication in the development team and the division of labour. This also means that the method must not be too formal, especially not in the creative phases.
Given these demands, we decided to develop a new method, called 'PROFIL' (PROduction strategy For Interactive Learning systems). The method is used for the development of several dozens of computer programs and has been slightly adapted over the years (Koper, 1989a; 1990; 1992). In this article a state-of-the-art description of the method will be given.

2.PROFIL

PROFIL has six phases: preliminary research, definition phase, script phase, technical realisation phase, implementation phase and the exploitation phase (figure 1).

PROFIL phases	*outcome*
1. preliminary investigation phase	course plan
2. definition phase	project plan per medium
3. script phase	script
4. technical realisation phase	master program
5. implementation phase	installed product
6. exploitation phase	summative evaluation

Figure 1. The phases and outcomes of each phase of PROFIL

These are the normal phases distinguished in educational technology literature and in project management. The difference is to be found in the phases itself. Because every phase ends with a formal document or product which - in the Ou - is submitted for a go-/no go decision at different organisational levels, the possibility for iteration between the phases is limited. Iterative processes are mainly to be found within the phases itself. Furthermore, the level of detail is worked out throughout the process. In the first three phases of PROFIL the same process is repeated at a different aggregation level. In the first phase the problem is at the course level. The outcome is the set of different medium applications the course will contain. Then for every medium application a separate development process must be started. The second phase is a further elaboration of the design for each application (here we will concentrate on the computer applications and not on written materials, tutoring or linear electronic media). The outcome is a project plan for every application. In the third phase the same process is repeated, but now in more detail. The result is a complete script for every aspect of the application. In every of the first three stages a prototype can be made if it is necessary, but it is needed at least in the script phase, before the complete script is finished. In the following sub-sections each phase will be discussed. The focus in this article will be on the design stages and less on technical realisation, implementation and exploitation.

2.1. Preliminary investigation

In the preliminary investigation phase the developers are concerned with the problem of the general design of a course. The outcome is (1) a definition of the different media applications which the course will contain and (2) a course plan which contains information about the development of the course from a management perspective. In the process the following interrelated aspects must be defined:

1. the objectives of the course
2. the target group characteristics (needs, prior knowledge, study skills, setting)
3. the didactic scenario of the course - based on a general didactic model - which specifies how the education has to be set-up in order to attain the objectives with the target group
4. which elements of the didactic scenario will be implemented in which physical media (e.g. individuals, texts, computers, video) available in the educational system.
5. For every medium application a draft technical design must be made. The level of detail must be enough to get rough estimates of development time and costs.

The idea is that a developer defines several serious alternative solutions which will then be compared. Every change in one or more of the five aspects, stated above, defines a new alternative solution for the course. After the generation of the different solutions, every alternative will be weighted in terms of (1) functionality, (2) incomes and costs, (3) feasibility, (4) compatibility with the characteristics of the educational system and with the policy of the institute. The choice of the alternative is made on the basis of the results of the judgement. This process integrates the problem of medium selection and is more fully worked out in Koper (1989b). The process of media selection is in PROFIL not limited to the weighting of different physical media, as in the media selection models of Anderson (1976), Gagné and Briggs (1979); Reiser & Gagné (1983) and Romiszowski (1988), but places emphasis on the generating, weighting and selection of the different media applications which can make up a course.
A small example will make this process more clear. One of our course projects in the faculty of law is to learn students the practical skills of a lawyer in law court. We considered several alternative solutions for the set-up of the course, to name a few (with an accent on differences in the physical media in which the didactic scenario is implemented):
1. a simulation of a law court in a real setting with actors, as is done at some other universities;
2. a computer program (CD-rom or interactive video) with a simulation of a law court, with or without motion video;
3. a case approach only with written materials or with some additional video materials;
4. not include the practical skill into the curriculum.

Every alternative is seriously considered in terms of functionality, feasibility and compatibility with the characteristics of the educational system and with the policy of the institute. The first option for instance is rather expensive as compared to the second option when the number of students, investment costs, travelling expenses and operational costs are taken into account. With the third option there is a doubt whether the intended learning objectives will be obtained. The fourth option is a more fundamental consideration, related to the goals of the curriculum. For us, the outcome of the process was that we have chosen option 2.

The preliminary research ends with the production of a 'course plan', in which the outcomes of the selection process are described. As is stated before, the activities in the definition phase are a repetition of the preliminary investigation at a different aggregation level (the media application level as compared to the course level). The different activities will be discussed in more detail in the definition phase.

2.2. Definition phase

In this phase the different media applications are analysed further with the objective of establishing a definition of the didactic and technical structure of the systems to be designed. The resulting definitions form the basis of the script which is a more worked out version of the product definition. The outcome of the definition phase is a project plan per medium application, in which costs and planning are worked out in more detail than they were in the previous phase. The definition process for courseware comprises two stages each of which has five steps (figure 2).

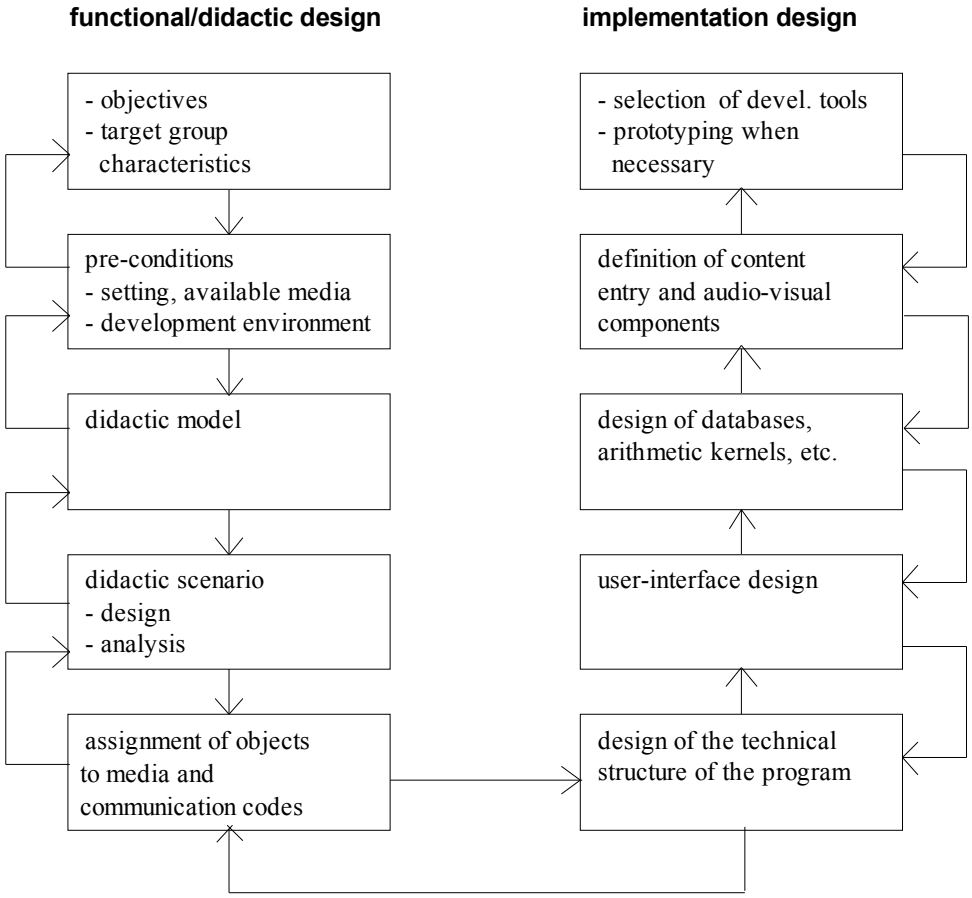

Figure 2. The definition process

The dotted lines in the figure suggest a feedback process in searching for an optimal solution. We shall now briefly discuss the different steps in the definition process for courseware development.

Objectives and target group characteristics
The definition begins with a careful analysis of the objectives that are to be achieved by the courseware package to be designed. The target group characteristics are for a great part derived from the preliminary research, but refined where necessary.

Pre-conditions: setting, available media and development environment
In this step the pre-conditions for the development of the courseware package are analysed. First the educational setting is analysed in terms of the system characteristics (general didactic approach, educational level, numbers of students, etc.) and the media which are available in the system. Three types of media can be distinguished: physical media, distribution media and communication codes. Physical media can be defined as the available 'hardware'. This includes the specific characteristics of the system which are defined by the system software, the type of user-interface for input and output and the devices. A computer running under MS-windows has other characteristics than a computer running under DOS or System 7. These are considered as different types of physical media.
In our definition the physical media do not only include standard media such as text books and electronic media but also persons (e.g. teachers or fellow students) which are available in the educational environment.
Distribution media are the carriers of the software. For example: paper, cd (audio, cd-rom or

cd-i), video cassette, diskette and laser disk.
Communication codes are the symbols in which the messages which form the communication are encoded. Examples of communication codes are: figures, tables, speech, text, music and moving pictures.
In this phase a list must be made of all the media (in the three definitions) which are available in the system. For courseware this means:

- a list of the physical media: the different types of computers (classical PC's, multimedia PC's, CD-i players, telematic facilities) available in the educational system and the 'surrounding' media, such as persons, audio-visual media and texts.
- a list of the distribution media which are available for the computer (cd-rom, diskettes) and for the surrounding media.
- a list of the (tele-)communication codes and facilities provided by the different computers and surrounding media. A multimedia PC - for instance - can display text, sound, still video, moving pictures (animated or video) and graphics. The input is tactile (via mouse, keyboard).

In this phase no selection is made yet for the implementation of the different objectives in media. Only the media which will be considered as candidate media in the selection process are listed.

The second activity in this phase is the definition of the development environment which is available for the production of the materials. In our institute we develop courseware for MS-windows at the moment with Toolbook, Turbo Pascal and some specific libraries and tools (e.g. graphical tools). Textbooks are developed in a Macintosh environment. The development environment sets the constrains for the implementation of the functional design later on in the development process.

Didactic model
The following step relates to the choice of a 'didactic model'. We define a didactic model as a theoretical construction, comprising variables and their relations, in which a relation is described between didactic functions, student characteristics and content matter. Reigeluth (1983) speaks about 'instructional-design theories and models'. Resnick (1983) distinguishes three components in an instructional theory. First, a theory about the specification of expert knowledge in a domain. Secondly a general learning theory and thirdly an instructional theory. A didactic model contains at least the third component: an instruction theory. Examples of didactic models include: mastery learning (Bloom, 1968), experiential learning (e.g. Dewey, 1938; Keeton, 1976; Kolb, 1984); the open education concept (Davies, 1980); apprenticeship learning (Resnick, 1988; Brown, Collins & Duguid, 1989); problem-based learning (Barrows & Tamblyn, 1980); programmed instruction (Skinner, 1968); the case method (Leenders & Erskine, 1989). Didactic models are more or less 'content-free'. The choice of a model will never be taken on the basis of a concrete objective itself, but on a class of which the objective is a part. The basic question for the designer is: which model is the best for achieving the objectives with the target group?
A didactic model does not have to be an existing model already described in the literature. A system of personal educational views of the developer can also be seen as a didactic model.
In many educational situations the didactic model is determined by a general educational policy. One of the advantages of the application of interactive media in education is that there is no necessary adhesion to a particular didactic model, but there is freedom of design.

Didactic scenario: design
A didactic scenario is designed on the basis of the didactic model chosen and the objectives to be achieved for a given target group. This means the design of a more or less 'ideal' learning environment in which the objectives can be achieved by the target group. The term 'ideal' refers to the fact that no account is taken of the implementation of the didactic scenario in the available physical media. A didactic scenario contains 'functional objects', i.e. objects of which only the functionality is of importance, the media or matter in which the functions are implemented can be changed latter on, although they are chosen as 'ideal' media. The consequence is that there is absolute freedom in the conception of all manner of objects. This gives the designer an enormous amount of imaginative freedom. The central objective is the development of a didactic scenario

that 'works', i.e. one which allows the set objectives for the target group to be achieved.
The development of a didactic scenario is, in practice, often a creative process in which the developer might ask himself the following questions:

- In what (ideal) learning environment could the objectives of the target group be achieved, given the principles of the chosen didactic model?
- What is the structure of the learning environment; what are the objects in the learning environment and what processes and activities take place?
- Are there other alternative learning environments imaginable and which solution is the better?

The choice of a didactic model and its elaboration for the objectives and the target group in a didactic scenario is mostly not linear, but interactive.

Didactic scenario: analysis
The didactic scenario is then analysed in terms of:

a. The structure of the learning environment, i.e. the comprising sub-environments en relations between the sub-environments. The result is a map of the learning environment.
b. The type of the functional objects in each sub-environment. The functional objects can be grouped in four types or classes: communication objects, tools, background objects and connection objects. Communication objects are the 'persons' in the scenario, e.g. a student, tutor, librarian or expert. Tools are all non-communication objects that can be manipulated by the student or by other communication objects in the scenario. Background objects are objects which cannot be manipulated, but set the context. Connection objects are objects which are needed to go from one sub-environment to another (e.g. 'doors', 'gateways', 'signs'). In an actual computer program there is also a fifth type of object: the controls, i.e. objects with which the application is controlled (e.g. menu bars).
c. The function of the different objects and between the different objects in the sub-environments. Important aspects of a functional description are: whether the object has a certain form of initiative; whether it is already present when the student enters the sub-environment, what the relations with other objects (especially the student) are and which information it contains.
d. The type of messages which are send between the various functional objects in the scenario. The type is bound to the communication modality, such as: written text, speech, visual information or tactile information.
e. The processes in the didactic scenario. The processes describe the order of activity and the involvement of the various objects. This can for example in the description of the phases. A more detailed description of the procedure occurs later in the scripts.

What is essential here is that a distinction is made between the design of functional objects and the design of implementation objects. In the didactic scenario only functional objects are designed. When a definite medium is chosen for the implementation of the functional object, then the design of the implementation objects starts. The latter are the objects a student will actually see in his learning environment. A functional object is in fact not more than a (often complex) set of functions, ordered under the name of the object. Speaking of a 'functional object' instead of separate functions provides an opportunity to group functions in natural units at a higher level of abstraction. This makes the design and the technical realisation easier.
For instance: the functional object 'teacher' in a certain scenario has a group of functions (e.g. explanation, questioning and feedback). The functional object 'teacher' can be implemented in a person or by other media, such as a computer or a textbook. If we refer to a "teacher" in terms of a functional object, we are exclusively concerned with the set of functions of the object in the scenario and not with the implementation media. A functional object can however never exist in reality without being implemented. It is however easier for the designer to think in terms of functional objects than in terms of more abstract, isolated functions.

For example:
If a didactic scenario says the following: "a student in a classroom is solving an arithmetical

problem stated in a book. He can use a note pad and a pen. The teacher looks over his shoulder and intervenes when he finds it necessary", than:

- The structure of the learning environment is: a classroom (only one sub-environment).
- The functional objects are: teacher, student, book with arithmetical problems, note pad and pen. The first two are communication objects and the latter ones are tools.
- The functions of the different objects are: Teacher: looks over the shoulder of student and intervenes when necessary. Student: solves an arithmetic problem, uses note pad and pen when necessary and listens to the teacher when he intervenes. Etc.
- The types of messages are: Teacher 'looks' (visual) at the activities of the student. The interventions of the teacher are of type 'speech'. Etc.
- As far as the processes are concerned: first the student receives the arithmetical problem in the book in one way or other (not specified), he solves it while the teacher observes him and speaks to him when necessary, when the student has solved the problem the process comes to an end.

The example is a reasonably 'everyday' didactic scenario. It may be that more abstract, less everyday and more imaginative scenarios can be developed to achieve the objectives with the selected target group. Perhaps it is important precisely in the design of programming for new implementation media, like computers, to develop entirely new didactic scenarios which are as effective as possible, but do not yet actually exist, in order to fully use the opportunities provided by the new implementation media and to discover new forms of application.

Assignment of functional objects to physical media and communication codes

In this stage two questions have to be considered:

1. Which objects (or functions) of the didactic scenario will be implemented in which physical media and which distribution media will be used?
2. Which communication codes will be used, given the possibilities of the physical media chosen?

With respect to the first question: for every object we have to decide in which physical medium it is to be implemented and which delivery media are used. When we have decided to implement an object in the medium 'computer', we must also select the type of computer from the list of available computers (e.g. Macintosh, MS-windows machine, Multimedia PC, DOS machine or CD-I player) and the delivery media to be used. A choice for a computer and delivery medium made for the implementation of one object ought to be consistent with the choices made for other objects.

When developing a computer application the choice for the computer as medium is already made, but it is well worth the effort to reconsider the implementation of the different functions in the other media at hand in the course. Maybe it is cheaper or more effective to put the help functionality in a written manual, or to use a person for the implementation of the teacher functions instead of the computer program. If we take for instance the various objects from the previous example, and distribute them in a standard MS-Windows computer, then we could decide to implement all the functional objects into the computer program: the teacher, the note pad, the pen and the book. Another optional breakdown is that the role of the teacher is played by a (skilled) person and the computer plays the role of the note pad, the pen and the book. Another option is to give the student the role of teacher and let the computer play the student, note pad, pen and arithmetical problem (depending on the learning objectives). There are still further ways of breaking this down. The originally selected media mix may have to be adjusted because it makes more sense or would appear more practical in this phase to assign the role of a particular object to another medium.

With respect to the second question: when certain objects or functions are assigned to a specific medium, it must be decided how the message is coded, given the characteristics of the object and the medium at hand (e.g. will the teacher object in a computer program be implemented in video, speech or text).

For both questions, media selection and code selection, the four factors mentioned earlier are of importance in weighting the alternatives: functionality, costs, feasibility and compatibility with educational system and policy.

Design of the technical structure of the program

Now we have decided which functional objects will be implemented in which medium, we can make a design of the implementation of the functional objects in the different media. Here we will concentrate on the implementation of functional objects in a computer program. The first step in the 'implementation design' is to define the overall technical structure of the system, given the didactic scenario of the application (which is a part of the overall didactic scenario: only those objects which will be implemented in the computer program). This means that different technical modules have to be identified which can be designed and programmed relatively independent of each other. Examples of such modules are databases and arithmetic kernels. Furthermore, if a didactic scenario comprises different sub-environments which are relatively independent of each other, than every sub-environment can be envisaged as a separate technical module in the system. One of the considerations in the breakdown of the program is the possibility to re-use existing programs or parts of programs: modules must be defined in such a way that existing modules can be re-used.

When the different modules are identified, the interfaces between the modules have to be designed. For instance the syntax of the query language for databases. It could be that databases or persons will be accessed via a network. In that case the network connections have to be designed/defined.

User-interface design

The design of the user-interface is a topic in itself, which cannot be dealt with in detail in this article. The design of the user-interface is done in four steps:

1. the design of the macro user-interface (number of screens, which objects on which screen and the positions of the objects on the screen, definition of audio channels and input devices).
2. the design of the meso user-interface (the functionality per object, the content of menu bars, specific controls, etc.)
3. the design of the micro user-interface (the exact functionality of the basic objects as buttons, list boxes, fields etc. work)
4. the graphical design of the user-interface.

As a starting point for the macro design of the user-interface we take the sub-environments and objects of the didactic scenario of the application. First, an implementation of the different sub-environments and their connections has to be established (e.g. a full-screen window for every sub-environment and the connection are made by choices in a menu bar). Secondly, all the objects per sub-environment have to be placed somewhere on the screen. Are they visible? Must they be opened by pushing a button? What communication code is used (text, audio, video)?

In our opinion the didactic scenario must be communicated to the student through the macro user-interface. This means that the different objects of the scenario must be designed in such a way that they are recognised as certain types of objects by the student (or by the graphical design, or with the help of a text in the heading, such as: calculator, expert, note pad, etc.). The consequence is that the macro design of the user-interface has to be 'derived' from the didactic scenario.

In the design of the meso user-interface, the designer fills in the different objects, according to the specified functionality of the objects.

The design of the micro user-interface isn't a part of most projects. When the Windows or Macintosh environment is used, the functionality of most micro objects are pre-defined (buttons, list boxes, etc.).

At the Ou we have defined standard user-interface implementation guidelines, for the macro, meso and micro level. This makes the translation of the didactic scenario into a screen solution rather straight forward.

Design of databases, arithmetic kernels, etc.

When databases or arithmetic kernels are part of the application, they have to be designed. For databases this means: the records and fields per database and the relationships between databases. Furthermore the technical implementation has to be thought-out (dbase or another format)?

For arithmetic kernels this means that they have to be defined (or bought) and also here the technical aspects have to be thought-out.

Definition of content entry and audio-visual components
One of the problems with courseware design is the entry of content matter in different formats: text, audio, video, formulas, etc. For that reason, it is a good practice to give attention to this topic as integrated part of the design methodology. One of the major considerations in this phase is whether one should design a special data-entry program or not. Text data can be entered in different ways:

- the educational technologists or programmer is to be trained in the subject matter and enters it himself. An alternative is that a content matter specialist puts the content on paper in a rather informal way. This is translated to the format needed for the computer program by the educational technologist or programmer.
- the content matter specialist specifies the content on paper in a pre-defined format (e.g. inscript, Koper, 1991). The programmer translates this paper work to code.
- the content matter specialist enters the content in a text processor (or other graphical tool) in a pre-defined format which can be read by a computer program.
- the content matter specialist enters the content in a special data entry program. This program must be made (or adapted) per project.
- the content matter specialist enters the content in the program (often called a shell) itself which has special features for data-entry.

There must also be attention for the problem of error correction when the data is already in the program. In a lot of situations, students, content matter specialists and other team members give lists of errors to the programmer and the programmer will correct it in the program. For this situation tag codes have to visible in the alfa and beta test phases. When provisions are made for data entry into the program, the content matter specialists and educational technologists can make the changes themselves.
Formula's can be entered in spreadsheets, simulation programs, Pascal code, or be specified on paper. With audio and video, the problem is a little more complex. Once the audio or video is recorded, it cannot easily be adapted, so good audio-visual scripts and standard screen formats have to be designed.

Selection of development tools and prototyping when necessary
In this step a selection is made among the available programming tools in the development environment for the concrete implementation of the design. With these tools it is possible to make a prototype. Prototypes serves several functions. The major ones are: (1) to test the feasibility of (aspects of) the project, (2) to get the opportunity to test the program in an early stage and (3) to test the communication in the project team (is this the result we wanted?).
Whether prototyping is necessary or not is on the one hand a question of budget and planning and on the other hand related to the complexity of the project. We do not prefer to make prototypes in the definition phase of the project, but later in the script phase. It is however possible and sometimes necessary to make a restricted prototype, especially when formats have to be defined or when the performance or feasibility is questioned. In our view prototyping must be done in about 10% of the planned programming time. Furthermore the prototype has to be as complete as possible on the user-interface side, because that gives a full view on the intended result.

2.3. Script phase

The script phase is an exact repetition of the former phase, but now in more detail. This means that the same steps are followed, but everything is now worked out fully. The level of detail which is needed depends on the specific setting, but it must match the needs of the production staff (programmers, AV producers/directors) to accomplish their work. In this phase we prefer to make a prototype. The outcome of this phase is a mixture of paperwork (text specifications, AV-scripts, graphics, etc.) and prototype(s).

2.4. Technical realisation phase

In the technical realisation phase the following steps will be taken:

- audio-visual materials are recorded
- a technical design of the program is made by the programmer
- an alfa version of the program is programmed
- the content is entered
- the graphical design of the alfa version is completed
- the alfa version of the program is tested by the different members of the project team
- a beta version of the program is produced
- the beta version is tested with students
- the final version (1.0) of the program is produced

The technical realisation of the product can be supported with a great number of methods, techniques and tools. In our situation, we use special libraries with standard objects and routines for the programmer, data-entry tools, test sides, automatic logging en questionnaires for testing purposes and tools for the graphical designer.

2.5. Implementation phase

In this phase the product is installed at the users. The most important aspect at the implementation is that the systems developed are available for users and that they accept the developed system and use it as it is intended. This means that the system must be copied; user instructions and training programs must be developed; recruitment activities must be set up; the package must be sold and/or distributed; managers must be informed/trained; etc.

2.6. Exploitation phase

In the exploitation phase users are supported by the use of the system; small problems are solved; the use is evaluated and the user situation is optimised. If, on the evaluation it appears that the system has to be thoroughly adjusted, this can lead to an revision plan, which is the start of a new cycle (next is the definition phase).

3.Conclusion

The systematic use of PROFIL solved the problems, mentioned in the introduction, we had with the design of courseware. However, there are still some aspects which needs further elaboration. One aspect is that it is necessary to develop criteria for the description of didactic models. As a consequence of the lack of these criteria it is not possible to define the relationship between didactic scenario's and didactic models strictly enough. Another aspect is that the cost-factor of the design needs some further consideration. We need more specific information about the way how to calculate the costs of alternative media and communication codes.
A further problem we have is with the selection of communication codes in terms of functionality factor, especially in multimedia applications: when do you use speech, when full screen, full motion video, when stills, when text? The literature on this aspect is still to vague.

At this moment we are finishing the development of a case-tool, called 'SHOC-tool' which supports working with PROFIL in the functional design aspects (the steps in the left side of figure 2). Furthermore we are working on the definition and integration of flexible standard objects on the different levels (functional design objects, implementation design objects and programmed objects at the macro, micro and meso level). The idea is that a designer must chose his functional objects from a set of standard objects and adapt them when necessary. Only when there isn't a suitable standard object at hand, the design of a new object will be considered.

References

Alessi, S.M. & Trollip, S.R. (1985). *Computer-based Instruction: Methods and Development,*

Englewood Cliffs, N.J.: Prentice-Hall.
Anderson, R.H. (1976). *Selecting and developing media for instruction*. New York: Van Nostrand Reinhold.
Barrows, H.S. & Tamblyn, R.M. (1980). *Problem-based learning: an approach to medical education.* New York: Springer.
Bloom, B.S. (1968). Learning for Mastery. *UCLA evaluation comment, 1 (2).*
Bork, A. (1984). Producing computer based learning material at the educational technology centre, *Journal of Computer based instruction, 11* (3),78-81.
Brown, J.S., Collins, A. & Duguid, P. (1989). Situated Cognition and the Culture of Learning. *Educational Researcher, 18(1),* 32-42.
Burke,R.L. (1982). *CAI sourcebook.* New Jersey: Prentice Hall.
Coad, P. & Yourdon, E. (1991). *Object-oriented Design.* Englewood Cliffs, NJ, Yourdon Press (Prentice Hall Building).
Davies, W.J.K. (1980). *Alternatives to class teaching in schools and colleges*. Counsel for educational technology, London.
Dewey, J. (1938). *Experience and education*. Phi Delta Kappan.
Eilers, H.B. (1979). *Systeemontwikkeling volgens SDM. [system development according to SDM].* Den Haag: Academic Service.
Gagné, R.M. & Briggs, L.J. (1979). *Principles of instructional design (2e ed.)*. Holt, Rinehart and Winston.
Hartemink, F.J.A. (1987). *Handleiding voor de Ontwikkeling van Educatieve Programmatuur*. PMI-reeks, nr.12, Enschede: COI.
Keeton, M.T.,(ed), (1976) *experiential learning: rationale, characteristics and assessment*, San Francisco: Jossey-Bass
Kolb, D.A. (1984). *Experiential Learning.*New Jersey: Prentice Hall.
Koper, E.J.R (1989a). *Inscript: een scripttaal voor het systematisch ontwerp van interactieve leersystemen. [inscript: a script language for the systematic design of interactive learning systems].* OTIC research rapport no. 4. Heerlen: OTIC, Open univeristeit.
Koper, E.J.R. (1989b). Een keuzestrategie voor de inzet van (electronische) media in het hoger onderwijs. [a strategy for the selection of (electronic) media in higher education]. *Tijdschrift voor Hoger Onderwijs,7* (3), 78-88.
Koper, E.J.R. (1990). D*e ontwikkeling van computerondersteund onderwijs aan de Open universiteit. [the development of computer based instruction at the Open university]*. COP. Heerlen: Open universiteit.
Koper, E.J.R. (1991). Inscript: a courseware specification language. *Computers & Education,16* (2), 185-196.
Koper, E.J.R. (1992). *Studieondersteuning met behulp van de computer [study support by means of a computer].* Utrecht: Lemma.
Leenders, M.R. & Erskine, J.A. (1989). *Case research: the case writing process* (3e ed.), University of Western Ontario, London, Ontario.
Reigeluth, Ch.M. (ed.). (1983). *Instructional design theories and models: an overview of their current status*. Hillsdale, NJ: Lawrence Erlbaum Associates.
Reiser, R.A. & Gagné, R.M. (1983). *Selecting media for instruction*. Englewood Cliffs, New Jersey: Educational technology publications.
Resnick, L.B. (1983). Toward a theory of instruction. In: S.G. Paris,G.M. Olson, H.W. Stevenson. *Learning and motivation in the classroom*. Hillsdale, New Yersey: Lauwrence Erlbaum Associates.
Resnick, L.B. (1988). Learning in school and out. *Educational researcher*, 16 (a), 13-20.
Roblyer, M.D. (1988). Fundamental Problems and Principles of Designing Effective Courseware. In D.H. Jonassen (Ed.), *Instructional designs for micro-computer courseware*(pp. 7-33). Hillsdale, N.J.: Lawrence Erlbaum Associates.
Romiszowski, A.J. (1988). *The selection and use of instructional media*. London: Kogan Page.
Skinner, B.F. (1968). *The Technology of Teaching.* New Jersey: Prentice Hall.
Wager, W. & Gagné, R.M. (1988). Designing computer-aided instruction. In D.H. Jonassen (Ed.), *Instructional designs for micro-computer courseware* (pp. 35-60). Hillsdale, N.J.: Lawrence Erlbaum Associates.

Yourdon, E. & Constatine, L.L. (1978). *Structured Design* (2e ed.). New York: Yourdon Press.